# Orientation Tracking with Quaternion Optimization

Jay Paek
*Department of Electrical and Computer Engineering*
*University of California, San Diego*
La Jolla, California
jpaek@ucsd.edu

***Abstract*—This project presents a novel approach to estimate the orientation of an Inertial Measurement Unit (IMU) using accelerometer data, optimizing it with motion models to enhance accuracy. Leveraging a projected gradient descent algorithms, we mitigate drift and noise, refining the estimation through motion dynamics optimization. Subsequently, the optimized orientations are utilized to stitch together panoramic images captured by a camera attached to the IMU, ensuring seamless transitions between frames. The methodology offers real-time orientation estimation, robust accuracy, and versatile applications in augmented reality, robotics, and navigation systems. Experimental validation demonstrates the effectiveness of the proposed approach in dynamic systems.**

***Index Terms*—Optimization, robotics, orientation tracking, sensor fusion, quaternions, projected gradient descent**

## I. Introduction

In the realm of robotics, the ability to accurately track the orientation of a moving body is of paramount importance for various applications ranging from navigation to object manipulation. In this project, we delve into the fundamental task of orientation tracking using data acquired from an inertial measurement unit (IMU) and ground-truth measurements provided by a VICON motion capture system.

The project is divided into two main components: orientation tracking and panoramic image construction. In the first part, we focus on developing algorithms to estimate the orientation trajectory of the rotating body using IMU measurements. This involves:

- Calibrating the IMU sensors
- Processing the IMU data to a desired format
- Producing initial predictions via the angular velocity motion model.
- Formulating an optimization problem to reconcile IMU measurements with the expected motion dynamics
- Implementing a projected gradient descent algorithm to optimize quaternion trajectories.

In the second part of the project, we extend our exploration to construct panoramic images by stitching together camera images captured by the rotating body. This component not only demonstrates the practical application of orientation tracking in real-world scenarios but also highlights the importance of sensor fusion in robotics applications. Through this project, we aim to gain insights into the challenges and methodologies involved in sensing and estimation tasks in robotics.

## II. Notations and Preliminaries

Let the orientation of our agent begin with the camera facing in the positive $x$-direction, and the positive $y$-direction and positive $z$-direction are placed so that $\vec{x} \times \vec{y} = \vec{z}$. The euler angles of a rotation are roll, pitch, and yaw, which are the counterclockwise rotation of the $x$, $y$, and $z$ axis respectively with respect to the right-hand rule.

In this project, we will be working in the space of 4 dimensional vectors known as quaternions, which are in the form $\mathbf{q} = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$, where $a, b, c, d \in \mathbb{R}$. Throughout this project, quaternions will also be present in a row vector form $[q_s, \mathbf{q}_v]$ where $q_s \in \mathbb{R}$ denotes the real number in the quaternion while the components of $\mathbf{q}_v \in \mathbb{R}^3$ denotes the scalar multiples of $\mathbf{i}, \mathbf{j}, \mathbf{k}$ respectively.

Addition and subtraction in the space will be the component-wise addition and subtraction. Let $.\circ. : \mathbb{H}\times\mathbb{H} \to \mathbb{H}$ define the usual multiplication in this space, and under this multiplication, the space produces a unique definition exponentiation. Define $\|.\|_2 : \mathbb{H} \to \mathbb{R}$ as the norm of any quaternion such that

$$\|\mathbf{q}\|_2 := \sqrt{a^2 + b^2 + c^2 + d^2}$$

Let $\mathbb{H}_* := \{\mathbf{q} \in \mathbb{H} : \|\mathbf{q}\|_2 = 1\}$ be the unit norm quaternion space. We will be using $\mathbb{H}_*$ to represent the orientation of the IMU data due do its closed-ness under multiplication, computational simplicity, convenient mapping from different orientations. Utilizing standard Euler angles can lead to a problem known as gimbal lock, which allows infinitely many combination of rolls, pitches, and yaws.

We will begin working with time $t = 0, 1, \ldots, T$, where $T$ denotes the index of the last IMU reading.

Let $\mathbf{q}_i \in \mathbb{H}_*$ denote the orientation of the IMU at the $i$th time step. $\tau_i, i \in \mathbb{N}$ is the change in time in seconds from time step $i-1$ to $i$. $\boldsymbol{\omega}_i \in \mathbb{R}^3$ is the angular velocity in radians per second measured by the IMU at time step $i$. $\mathbf{a}_t \in \mathbb{R}^3$ is the acceleration vector at time step $t$.

## III. Problem Formulation

With initial orientation quaternion $\mathbf{q}_0 = 1$, we can solve for $\mathbf{q}_{t+1}$ for any given $\mathbf{q}_t$ by $k : \mathbb{H}_* \times \mathbb{R}^3 \to \mathbb{H}_*$ under its defined multiplication and exponentiation:

$$\mathbf{q}_{t+1} = b(\mathbf{q}_t, \frac{\tau_t}{2}\boldsymbol{\omega}_t) := \tau_t \mathbf{q}_t \circ \exp([0, \frac{\tau_t}{2}\boldsymbol{\omega}_t])$$

Which is integrates the angular velocity over time, then applies the instantaneous rotation to the current orientation.

In addition to the kinematics model, we want to apply the accelerometer data in order to make sure that the rotation measurements preserve the intrinsic measure of gravity. With initial gravity quaternion $\mathbf{a}_0 = [0, [0, 0, -1]^T]$, we will calculate the intrinsic gravity under the angular kinematics model at $i$th time step by applying an algebraic conjugation on the initial acceleration by $\mathbf{q}_t$ i.e. $a(\mathbf{q}_t) := \mathbf{q}_t^{-1} \circ \mathbf{a}_0 \circ \mathbf{q}_t$.

Now we have a two metrics: one of which measures how well the orientation obeys the angular kinematics and the other which measures how well the orientations estimate the gravity. The goal is to rely on the accelerometer data because it is the best predictor of orientation. We can compensate some precision in the kinematics model in order to perfect the prediction of intrinsic gravity. We can formulate two different cost functions for different priorities: convenience, and robustness.

Our second metric is easy to model. We want to make sure the gravity under the rotations matches with the gravity read by the accelerometer. Therefore, we want to minimize the following cost function $c_2 : \mathbb{H}_*^T \to \mathbb{R}$, a function that maps from $T$ unit quaternions to a real number defined as follows:

$$c_2(\mathbf{q}_{1:T}) := \sum_{i=1}^{T} \|\mathbf{a}_i - a(\mathbf{q}_i)\|_2^2$$

To formulate the the cost for the kinematics model we can either take a naïve approach or a more refined approach. For the naïve interpretation of quaternions, we can just find the Euclidean distance between the given $\mathbf{q}_{i+1}$ and $b(\mathbf{q}_i, \frac{\tau_i}{2}\boldsymbol{\omega}_i)$. Define $c_1 : \mathbb{H}_*^T \to \mathbb{R}$.

$$c_1(\mathbf{q}_{1:T}) := \sum_{i=1}^{T-1} \left\| \mathbf{q}_{i+1} - b(\mathbf{q}_i, \frac{\tau_i}{2}\boldsymbol{\omega}_i) \right\|_2^2$$

Alternatively, we can consider the fact that the additive inverse of any quaternion will also represent the same rotation. Therefore, we do not want to penalize such values, which $c_1$ indeed does. Define $c_1' : \mathbb{H}_*^T \to \mathbb{R}$.

$$c_1'(\mathbf{q}_{1:T}) := \sum_{i=1}^{T-1} \left| \left\| \mathbf{q}_{i+1} - b(\mathbf{q}_i, \frac{\tau_i}{2}\boldsymbol{\omega}_i) \right\|_2^2 + \left\| \mathbf{q}_{i+1} + b(\mathbf{q}_i, \frac{\tau_i}{2}\boldsymbol{\omega}_i) \right\|_2^2 - 2 \right|$$

This formulation comes from the triangle inequality where we are taking summing the distances of $\mathbf{q}_{i+1}$ to $b(\mathbf{q}_i, \frac{\tau_i}{2}\boldsymbol{\omega}_i)$ and $b(\mathbf{q}_i, \frac{\tau_i}{2}\boldsymbol{\omega}_i)$ to $\mathbf{q}_{i+1}$. We want this to be as close to the distance from $\mathbf{q}_{i+1}$ to $-\mathbf{q}_{i+1}$, which is just 2. This will allow the quaternion to freely choose whether to be closer to either representation of the same orientation.

Construct two cost function $f, f' : \mathbb{H}_*^T \to \mathbb{R}$ formulated as follows:

$$f(\mathbf{q}_{1:T}) := c_1(\mathbf{q}_{1:T}) + c_2(\mathbf{q}_{1:T})$$

$$f'(\mathbf{q}_{1:T}) := c_1'(\mathbf{q}_{1:T}) + c_2(\mathbf{q}_{1:T})$$

## IV. Technical Approach

Before any computation, it is crucial to preprocess the IMU data because the incoming data are integer values that corresponding to the voltage readings on the accelerometer and gyroscope. During the time where the IMU is static, we must find the resting state values to calibrate the IMU. Configure the resting state values to 0 (other than the $z$-axis acceleration), we can represent the initial state. Lastly, we need to scale the voltage reading so that the deviations represent the angular velocity in $\frac{radians}{s}$ and acceleration in $\frac{m}{s^2}$. Such scaling factors can be found in the device documentation.

Our solution is initialized from the kinematics motion model from the angular velocity reading of the IMU, as defined in $b$. Given the two cost functions $f$ and $f'$, we will apply gradient descent with 15 iterations and and a step ratio of 0.2 for stable convergence. Through Python, we will compute and model the quaternion orientations and optimize the orientations via Jax, a high performance array computing package that will find the gradient for the cost function. After taking a gradient step scaled by the step ratio, we will normalize each quaternion since $\mathbb{H}_*$ is not closed under addition.

Next, using the optimized orientations, we will stitch together a panorama by taking the following steps:

- Determine pixel-to-degree ratio
- Extract Euler angles from quaternion for time $t$ with image.
- Rotate image based on roll to reverse camera rotation
- For each pixel in the image, place coordinate onto sphere with radius 1 using pitch and yaw, where to positive $x$ direction is considered to be (0,0).
- Project pixel coordinate on sphere to cylinder of height 1000 pixels and diameter 1000 pixels
- Color the coordinate and some of its surrounding pixels with the corresponding pixel value. This is because we are projecting onto a larger space, so the projection mapping may leave empty pixels between two adjacent pixel in a given image.

The longitudinal translation and projection of a pixel directly stems from the yaw of the orientation, which makes determining the $x$ coordinate of the pixel very easy. For $y$, the projection becomes obvious after some inspection. Due to the sheer volume of pictures, not all will be processed for utilization, but still take at least 150 frames to ensure

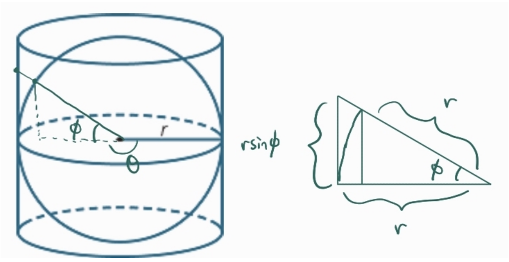

Fig. 1: Using the pitch $\phi$ in sphereical coordinates to project pixel to the enclosing cylinder

smooth transition between frames. For this particular project, the pictures are assumed to have height of 40 degrees and width of 60 degrees on a sphere.

## V. Results

The proposed solution performed well under the datasets from Professor Nikolay Atanasov with runtimes for the optimization to take under 5 seconds.

Both cost functions converged at around the same rate. Additionally, they did not have significant difference when predicting the gravity vector and generating the panorama. It seems that the second cost function does not provide a noticable advantage.

The following figures will only display the

- Ground truth roll, pitch, and yaw values from VICON data (if given)
- Initial and optimized roll, pitch, and yaw estimates
- Initial and optimized acceleration vector estimates
- Stitched panoramas generated under ground truth VICON data (if given) and panorama generated under orientation estimates

for the datasets with camera data (datasets 1,2,8,9,10,11). The orientation estimates for datasets 10 and 11 will not have a ground truth orientation plot since it is not given.

Despite a deviation from a more reasonable cost function by taking the quaternion $\log$ to see if the rotation for $\mathbf{q}_{t+1}$ perfectly reverses the orientation $b(\mathbf{q}_t, \frac{\tau_t}{2}\boldsymbol{\omega}_t)$, both cost functions seemed to converge to a good estimate.

Fig 1 and Fig 2 display the orientation and the gravity vector estimations for dataset 1 and 2. Fig 3 and Fig 4 are the panorama stitches generated from the corresponding orientations.

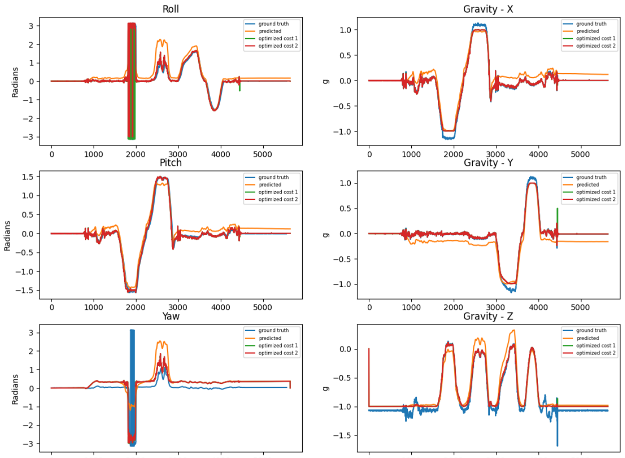

Fig. 2: Orientation (left) and gravity (right) predictions from ground truth (blue), inital prediction (orange), and optimized by $f, f'$ (green and red, respectively)

The plot highlight the fact that the quaternion trajectories were indeed altered to fit minimize the gravity error. The green and red plots are closer to the ground truth than the orange plot. It can be seen that there are more deviations in the orientation plots than the gravity plots. This is most likely because $c_2$ is orders of magnitudes larger than the first cost term.

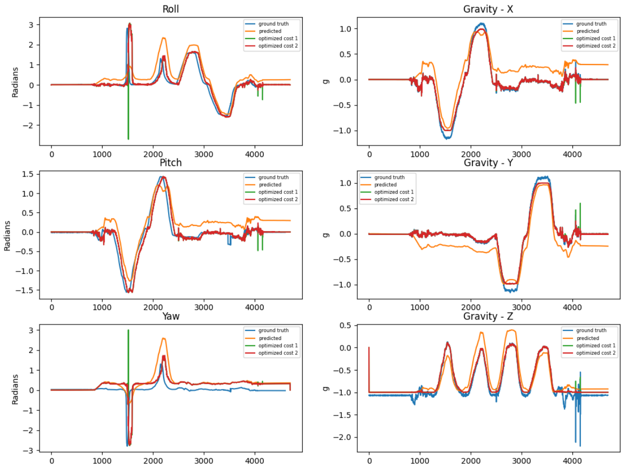

Fig. 3: Orientation (left) and gravity (right) predictions from ground truth (blue), inital prediction (orange), and optimized by $f, f'$ (green and red, respectively)

Another observation is that the green and red lines are identical almost everywhere. Since the cost functions' landscape are most likely to be similar.

The Fig. 4 and Fig. 5 are the panoramas stitched together given the orientations for dataset 1 and 2, respectively. A common occurrence across all of the panorama stitches is the center of the panorama being a little offset. Additionally, this often occurs when the camera leaves the initial direction it was facing and then returns to the same orientation.

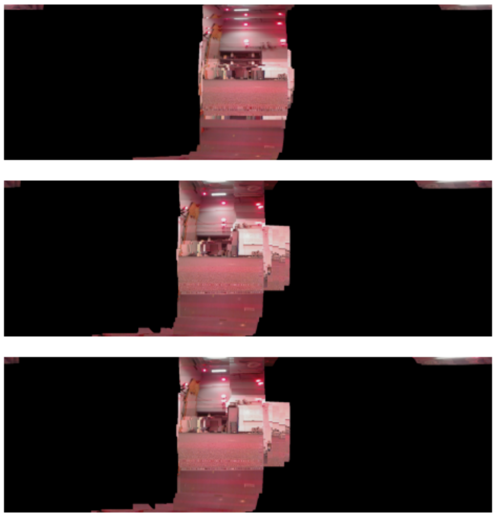

Fig. 4: (from top to bottom) Panorama stitches from VICON (ground truth) data, $f$-optimized, and $f'$-optimized orientations.

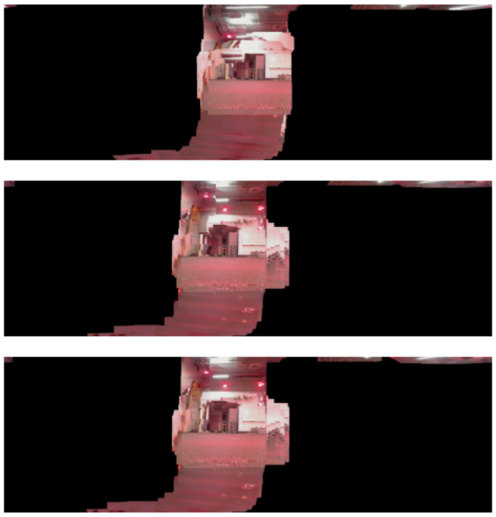

Fig. 5: (from top to bottom) Panorama stitches from VICON (ground truth) data, $f$-optimized, and $f'$-optimized orientations.

Fig. 6, 7, 8, 9 are the plots and panoramas for dataset 8 and 9. The quaternion initalization predicts orientation decently well but doesn't agree well with the acceleration. The optimizer performed well, but failed to optimize the last few times steps for the yaw. This caused the panorama to have a significant offset for the center, due to the overlapping feature of the panorama generator. This is inevitable since the error accumulates over time and motion, so uncertainty will unequivocally grow. However, disregarding the last few pictures that displace the center, we can see a very good reconstruction of the surroundings. Adding weight onto the gravity estimation error did not lead to better results. The cause is most likely to be the corresponding peaks of the acceleration vector which the cost function is try to optimize upon.

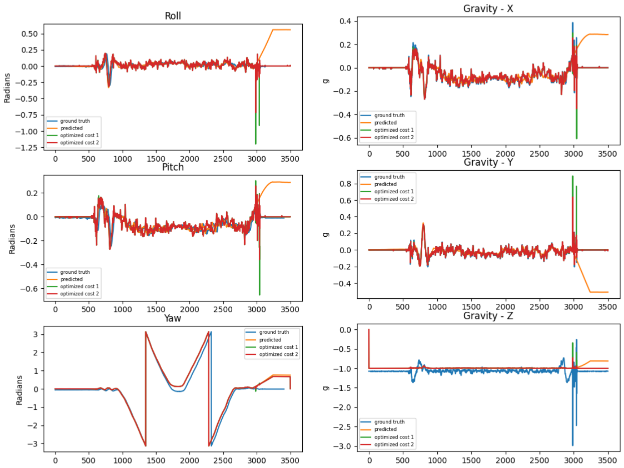

Fig. 6: Orientation (left) and gravity (right) predictions from ground truth (blue), inital prediction (orange), and optimized by $f$, $f'$ (green and red, respectively)

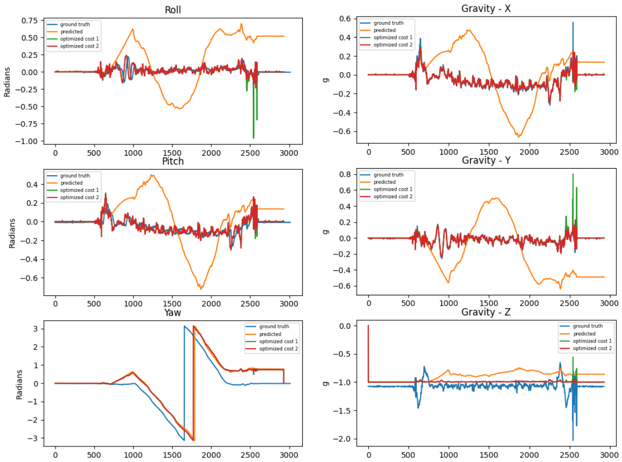

Fig. 7: Orientation (left) and gravity (right) predictions from ground truth (blue), inital prediction (orange), and optimized by $f$, $f'$ (green and red, respectively)

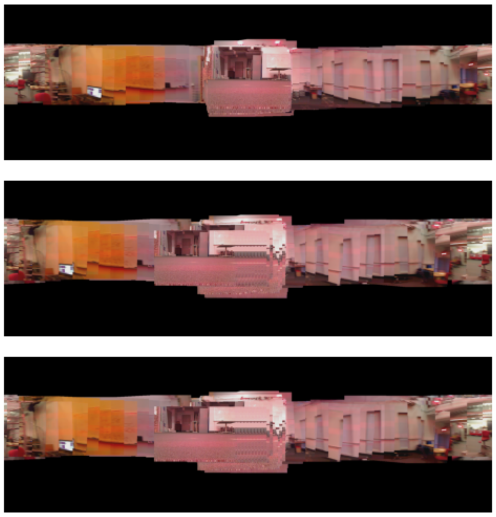

Fig. 8: (from top to bottom) Panorama stitches from VICON (ground truth) data, $f$-optimized, and $f'$-optimized orientations.

For the testing datasets, the only form of validation was the accelerometer readings and the panorama construction. Fig. 9, 10 show the performance on dataset 10 and Fig 11, 12 for dataset 11. The stitches of the panorama seem to align pretty well. However, dataset 11 faced a more scattered photos because the vast variations in pitch. There were only 160 pictures, so possibility of interpolation and smoother transition was impossible.

## Acknowledgment

I would like to acknowledge Jay Paek for outstanding work and commitment. I would also like to thank Professor Nikolay Atansov for accepting my late submission for code.

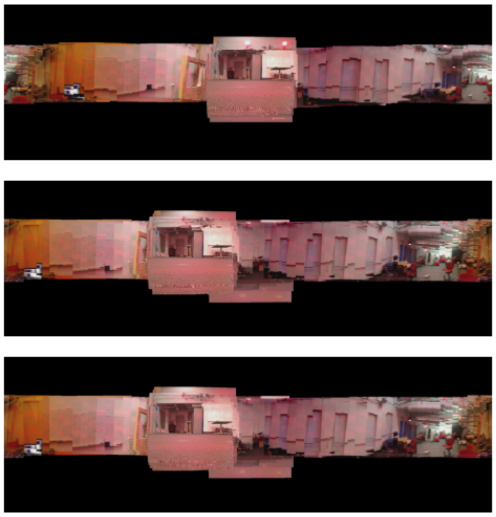

Fig. 9: (from top to bottom) Panorama stitches from VICON (ground truth) data, $f$-optimized, and $f'$-optimized orientations.

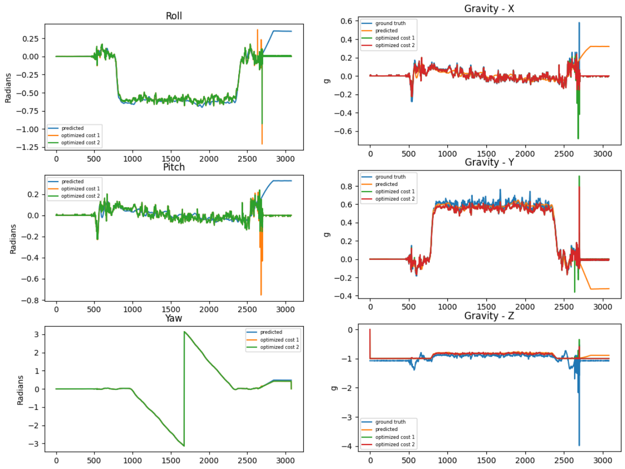

Fig. 10: Orientation (left) and gravity (right) predictions from inital prediction (blue), and optimized by $f, f'$ (orange and green, respectively)

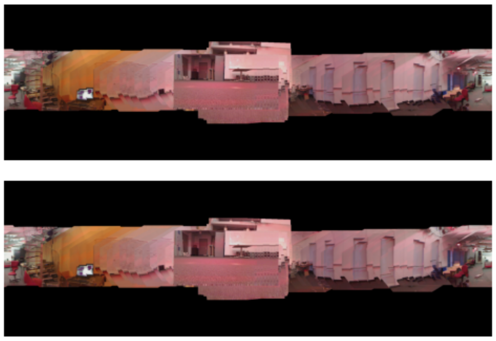

Fig. 11: (from top to bottom) Panorama stitches from $f$-optimized and $f'$-optimized orientations.

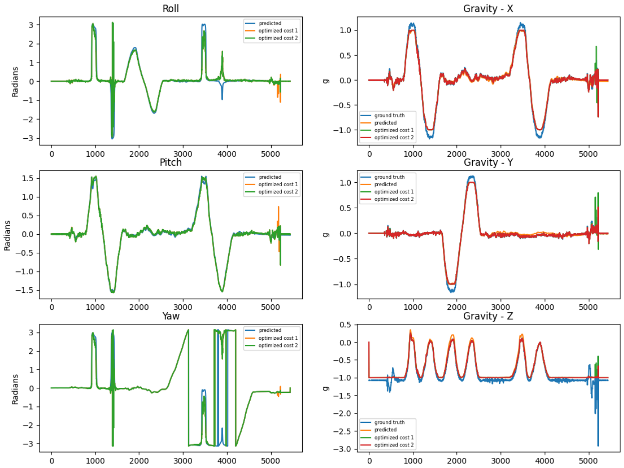

Fig. 12: Orientation (left) and gravity (right) predictions from inital prediction (blue), and optimized by $f, f'$ (orange and green, respectively)

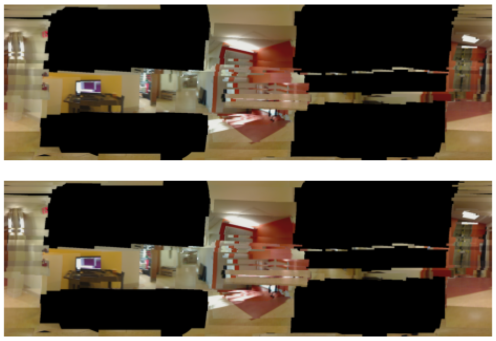

Fig. 13: (from top to bottom) Panorama stitches from $f$-optimized and $f'$-optimized orientations.